理研シリコン　取扱説明書

第 1 版（2009/05/07）

理研 BNL 研究センター　　外川 学

目次

1. はじめに 2
2. 用意するもの 3
3. セットアップ 5
IFC ボード–シリコン周り 5
PC、VME モジュールの設定 6
DAQ 7
4. 信号をオシロで見る～データ取得 9
オシロスコープで見る 9
データ収集 12
5. アナリシス 12

はじめに

本シリコン検出器は PHENIX VTX 検出器アップグレードプロジェクトのストリップ検出器部として、H14 年、当時理研の東城順治氏が BNL の Zeng Li 氏とで作った世界初の片面 2 層読み出しのシリコンストリップ検出器である。(参考資料 : Nucl. Instrum. Meth. A518 : 300-304, (2004)、Nucl. Instrum. Meth. A535 : 404-409, (2004)) .

読み出しは p 側で、元々80 μm*1000 μm のセル構造に 2 つの独立な p 電極が渦を巻いている。これをアルミ電極で一方向にとつなぎ合わせていき、ストリップ構造を作る。その際、ダブルメタル構造により、別方向への繋ぎ合わせを作ることにより、2 次元読み出しを実現する。本検出器では、斜め方向に接続している（u 方向、4.57° 。斜めに読む分一部不感領域有)。実質的には x 方向 80 μm ピッチ、y 方向 1000 μm ピッチで、有効面積およそ 3 cm× 3 cm の検出器となっている。読み出しチャンネル数は x:384, u:384 の計 768 である。

読み出しは IDEAS 社の VA2 バイキングチップを使って、VME モジュール,V551B によるクロックを用いて、シリアルに転送し、差動 amp 出力を ADC V550 モジュールにより読みだす。動作電圧やクロックなどの電圧変換、差動 amp 出力などはインターフェースボード(IFC)を通して行う。

VA2 : http://ccjsun.riken.go.jp/~togawa/RikenSilicon/va2.pdf

V551B : http://ccjsun.riken.go.jp/~togawa/RikenSilicon/caen_v551b.pdf

V550 : http://ccjsun.riken.go.jp/~togawa/RikenSilicon/caen_v550.pdf

本シリコン検出器は厚さ 400 μm、250 μm のものがある。KEK で行われたテスト実験の結果、検出効率は 400 μm のものはほぼ 100%であったが、250 μm のものに関しては、60% 程度と低い。これは S/N が悪い（ノイズは変わらないが、シグナルがほぼ半分）ことに起因している。リファレンス検出器として使用する場合（しかないと思うが)、400 μm のものが推奨される。また、0° 入射の際には、電荷の収量が x と u ストリップで偏りがでることも見られている。これはセル内の渦巻き状の p 電極ラインが、電荷収集の際の拡散効果と同程度の太さであることが起因していると考えている（新規モデルでは、p 電極を細くすることにより、この現象を抑えることに成功している)。以上の結果は JPS で紹介されています。

参考 JPS スライド：http://ccjsun.riken.go.jp/~togawa/RikenSilicon/JPS03March.pdf

用意するもの

- シリコン　400 μm (#3,4,9), 250 μm(#5,6,7,8) 使う分
  * 真空パックされてます（してないのもありますが。。）戻す時は真空パックしてください。
- サポートストラクチャー　シリコン台数による
- インターフェースボード (IFC) 1 台。何台かあるが、一番調整されている#1 を使用。
  * V551B との接続に必要な 20 ピンフラットケーブルも付いていると思います。
- ケーブル　（電源用 #1：2 本、 #2：7 本、#3：5 本、#4：3 本、白のコネクタ）
  （シグナル用　#1：2 本、#2：10 本、#3：7 本、#4：5 本、銀のコネクタ）
  * 一台読む分で、#1：1 本、#2：2 本、#4：1 本ずつ必要です。一台増える毎に#1：1 本、#3：1 本ずつ追加で必要になります。また、#2,3,4 は互換があり（おそらく CNS の物とも互換）、長さに合わせて代替可能です。#1 だけ特殊（ピン逆配置）になってます。複数枚接続は、
  http://ccjsun.riken.go.jp/~togawa/RikenSilicon/Cable20021015.pdf 参照。
- 電源（菊水±18V、2.5A のもの）と、IFC への接続ケーブル それぞれ 1 つ
- シリコン用電源、ORTEC 710 が使いやすい シリコン台数による 4 ch/module
- SHV-LEMO 変換　（ORTEC 710 を使う場合）シリコン台数による
  * シリコンの電源供給が LEMO 接続のため。空中配線は危険ですのでおやめください。また、本シリコンにはロードレジスター用の抵抗が入っていませんので、この変換の際に数 MΩ の抵抗を入れてください。（ノイズを抑えることができます）。また、コンデンサもかませてフィルターも作るといいと思います。時定数は適切に。
- VME クレート 1 台、モジュール（SBS620：1 台, V550：2 台, V551B：1 台）
  * 本来 V550 は一台で良いのですが、RESET シグナル用に一台使用します。本検出器の特殊仕様ということで。RESET シグナルを外部から提供する場合は一台でいいです。（詳細はセットアップのとこで）V550, V551B も放射線研にあります（AUX コネクタははずしてあります。ジャンパーピンもそのように設定してあります）。
- PC（SBS620 PCI を挿したもの）　1 台

以上の物品のほとんどが放射線研究室の２３７にあります。シリコンは黒いデシケータ内に入っています。他は入ってすぐの左の棚に置いてあります。SBS620(VME、PCI 共)と ORTEC 710 はこの検出器用に確保していません。竹谷氏などにご相談を。

図 1. 237 号室のシリコン置場
四日市氏管理のデシケータです。

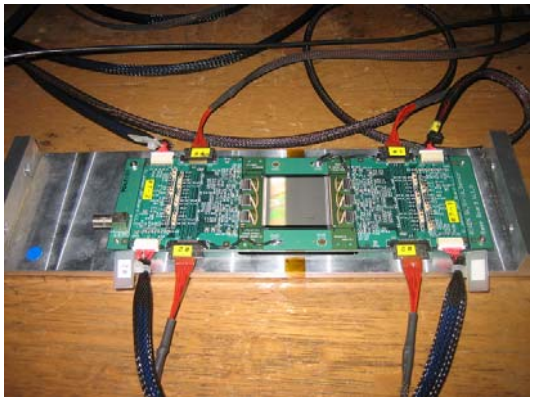


図 2. シリコン検出器、ケーブル（電源、シグナル）、サポートストラクチャー

図 3. 菊水電源(±18V, 2.5A), IFC ボード

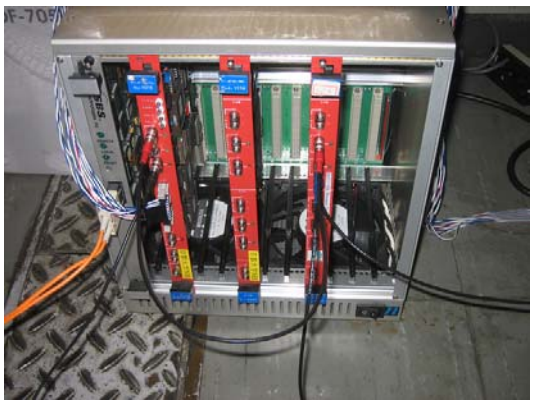


図 4. VME クレート、SBS620, V551B, V550×2 （向かって左から）

図 5. SHV to LEMO 変換。（これは京都の物。是非作ってください）

セットアップ

IFC ボード-シリコン周り

まずは電圧を設定してください。IFC ボードに供給する電圧は±15 V です。VA2 での動作電圧は+-2 V ですが、IFC ボードでこの値に変換され、供給されます。図 6 のように接続し、シリコンと接続する前に、IFC ボード上の電圧、avss, avdd, dvss, dvdd がそれぞれ、-2 V, +2 V, -2 V, +2 V になっていることを確認して下さい。(これらが VA チップに供給されます。はじめの a, d はアナログ用、デジタル用の電源を意味します)

次に図 6 のように IFC ボードとシリコン基板とを接続してください。1 台のみのシリコン側のケーブル接続は図 2 参照（既述していますが、#1 のみ逆接続です。間違えると電流値が異常になります)。電源供給の後、シリコン基板上で avss, avdd, dvss, dvdd が規定値であることを確認してください。また、菊水電源の方の電流値も正常か確認してください。参考値は以下の通り。

IFC のみで電圧を印加したとき　-15 V：-0.65 A　+15 V：0.65 A
一台シリコンをつないだ時　　-15 V：-1.00 A　+15 V：0.85 A

シリコンを 1 台ずつ増やすたびにこの差分が上乗せされます。また、これらの値は VA をオペレートすると±0.1 V 程度変化します。

以後、何台も使う場合は、#4 を#3 に変えて、次の#1 に#3 を接続し、シリアルに接続していく。最後 IFC ボードに帰る部分だけ#4 で。その際、#1 が入る部分の JP1 はショートさせなければならない。(基本はオープンになっています）半田でショートさせたり、オープンしたりするので、接続の際は最新の注意を行ってください。(2 台目以降はショートさせなくても、させても OK だった気がします。現在ショートしているものを 1 台目に持ってくるとよいと思います。)

正常に電圧がかかっているようであれば、次に VA チップの pre-amp, shaper に対しての設定を行います。（この可変抵抗は一度設定すればそこまでずれることはなく、既に設定すみなので、はじめは必要ないと思います。モジュールを動かして、DAQ で取ることが最優先です。本実験の前には確認して下さい。）設定は各チップ毎に pre_vias, sha_vias,vfp,vfs に対して行います。Nominal な値は VA チップの仕様書に、pre_vias, sha_vias は電流量で、vfs, vfp は電圧値として記載してあります。いずれもデジボルでテストピンを使って電圧値を測定し、その隣にある可変抵抗で調整します。*_vias は入れている抵抗から適切な電圧値を計算し、そのように調整してください。それぞれの電圧の基準は GND ではなく、avss、もしくは avdd になっているので、注意してください。

（http://ccjsun.riken.go.jp/~togawa/RikenSilicon/SmtSchematic20021114.pdf 参照）

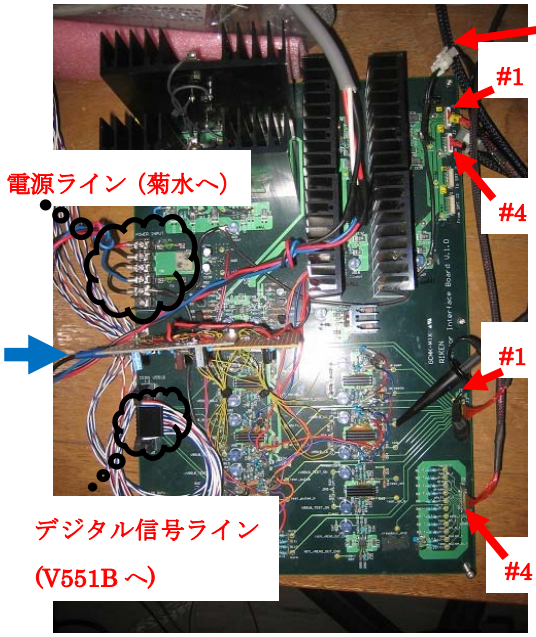


図 6　IFC ボード

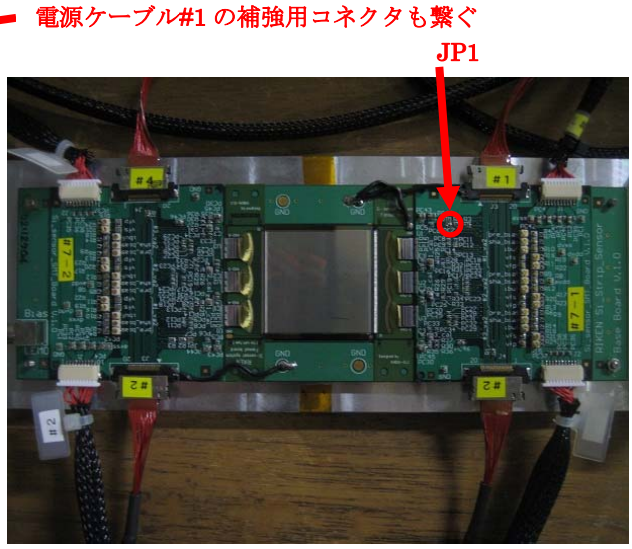


図 7 シリコン基板への接続。一台のみの読み出しは上記のように接続。2 台目以降ある場合はシリアルに、＃4 を#3 にし、次の#1 が#3 の部分に接続。以後同様に接続していき、最後 IFC に戻すものだけ#4 で。

PC、VME モジュールの設定

PC に VME PCI を挿し､VME クレートに挿した VME モジュールと接続してください。また、VME クレートに V550 を 2 台、V551B 1 台を挿し（挿す前にそれぞれのモジュールアドレスを確認してください。VME サイクルの際に必要です。ボード上のスイッチで設定しています。変更可です)、V551B とインターフェースボードをフラットケーブルで接続してください。電源をいれ、PC を通して VME に正しくアクセスできることを確認してください。（VME コマンドなどで V551 が動くかなど。以下にある初期化コマンドでアクセスランプが明滅すればおおよそ OK だと思います）

V550, V551B の初期化を実験構成に合わせて行います。主にチャンネル数やトリガーが来てからの各動作のタイミングを設定します。詳細は VA チップ、V551B の仕様書に記載してあります。設定例は DAQ の初期化プログラムを参考。

V550 には ADC のペデスタルレベル、ゲイン（レンジ）の設定をボード上で行います。大まかなレンジはジャンパーで切り替えて、細かな設定はスライダで設定します。デフォルトでどの設定にするか難しいので、DAQ でのデータ集がうまくいった後に、データを見ながら、全チャンネルのペデスタルが 0 以上になること、オーバーフロウがほとんどないこと、MIP が見えること、など実験に対する必要条件に合わせて設定してください。

DAQ

実験構成に合わせて作成してください。以下、nagidaq でのプログラムです。

初期化

主に割り込みの設定、V551B の出すクロック数（読み出しチャンネル数に相当）の設定、V550 の Zero suppression の設定などを行います。トリガーが来てから、VA チップに送るシークエンスのタイミング，T1~T5 もここでせっていします。シークエンスは FIG.4.3 (V551B の資料からそのままのタグ)参照。V550 の ADC では、Clock のパルス毎に VA チップからシグナルが出力され、それを Clock と同期した Convert の立下りでフラッシュしますので、主に重要なのはこのタイミングを決める T5 です。これは実験状況（ケーブルの長さ）で決まりますので、データ収集の際、オシロを見ながら決めてください。（オシロで見てみる参照)。

- T4 はクロックの周波数を決め、これが VA チップ読み出しの速さを決めます。本セットではここは 400kHz より遅くしてください。(どこかにそれより遅い時定数のコンデンサが絡んでるため、早い読み出しは前のシグナルが入り込んできます。）初期化の例の部分では 400kHz（2500ns）で設定してあります。

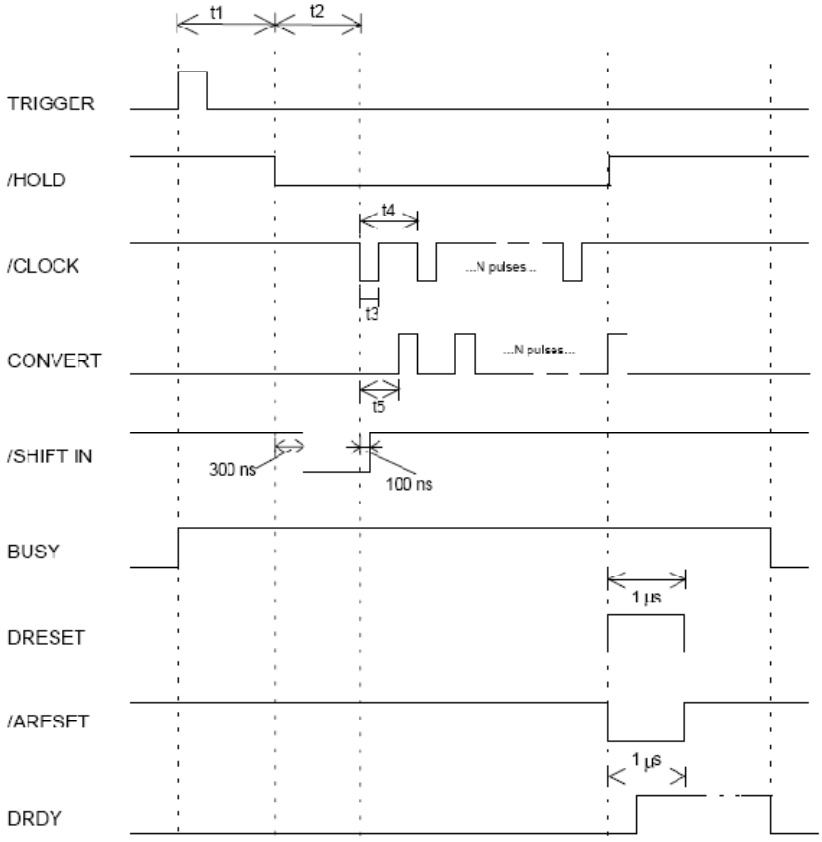


Fig. 4.3: Standard Operation Sequence

以下、初期化プログラムの例です。(VME アドレスは V551B:1111, V550:2222, 4444 です)LAM は 2 つ目の V550 で立てています。

```
//Initialize V551(Sequencer) for SSD"
//clear
VmeCycle( SBS620_WORD, SBS620_WRITE, SBS620_AM_09, 0x11110004 );
//test register normal mode
VmeCycle( SBS620_WORD, SBS620_WRITE, SBS620_AM_09, 0x1111000a );
//status reg 0
VmeCycle( SBS620_WORD, SBS620_WRITE, SBS620_AM_09, 0x11110008 );
//clock 768ch for reading 1 sensor
VmeCycle( SBS620_WORD, SBS620_WRITE, SBS620_AM_09, 0x1111000c, 0x0300 );
//clock 1536ch for reading 2 sensors
//  VmeCycle( SBS620_WORD, SBS620_WRITE, SBS620_AM_09, 0x1111000c, 0x0600 );

//internal DAC 0
VmeCycle( SBS620_WORD, SBS620_WRITE, SBS620_AM_09, 0x11110018 );
//int vec 0
VmeCycle( SBS620_WORD, SBS620_WRITE, SBS620_AM_09, 0x11110000 );
//int level 0
VmeCycle( SBS620_WORD, SBS620_WRITE, SBS620_AM_09, 0x11110002 );
//t1 550ns
VmeCycle( SBS620_WORD, SBS620_WRITE, SBS620_AM_09, 0x1111000e, 0x0005 );
//t2 450ns
VmeCycle( SBS620_WORD, SBS620_WRITE, SBS620_AM_09, 0x11110010, 0x0010 );
//t3 1240ns
VmeCycle( SBS620_WORD, SBS620_WRITE, SBS620_AM_09, 0x11110012, 0x003e );
//t4 2500ns
VmeCycle( SBS620_WORD, SBS620_WRITE, SBS620_AM_09, 0x11110014, 0x007c );
//t5 2340ns
VmeCycle( SBS620_WORD, SBS620_WRITE, SBS620_AM_09, 0x11110016, 0x002c );
//clear
VmeCycle( SBS620_WORD, SBS620_WRITE, SBS620_AM_09, 0x11110004 );
```

```
// Initialize V550(ADC) for SSD"
//clear
VmeCycle( SBS620_WORD, SBS620_WRITE, SBS620_AM_09, 0x22220006 );
// set channel num : 768ch for 1 sensor
VmeCycle( SBS620_WORD, SBS620_WRITE, SBS620_AM_09, 0x22220004, 0x0618 );
// set channel num : 1536ch for 2 sensors
//  VmeCycle( SBS620_WORD, SBS620_WRITE, SBS620_AM_09, 0x22220004, 0x0c30 );

//int level 0
VmeCycle( SBS620_WORD, SBS620_WRITE, SBS620_AM_09, 0x22220000 );
//at first status reg 0
VmeCycle( SBS620_WORD, SBS620_WRITE, SBS620_AM_09, 0x22220002 );

//set pedestal & threshold 0
for( int add = 0; add<0x2000; add+=4 ){
  VmeCycle( SBS620_LONG, SBS620_WRITE, SBS620_AM_09,
            0x22222000 + add, 0x00000000 );
  VmeCycle( SBS620_LONG, SBS620_WRITE, SBS620_AM_09,
            0x22224000 + add, 0x00000000 );
}
//status reg 2
VmeCycle( SBS620_WORD, SBS620_WRITE, SBS620_AM_09, 0x22220002, 0x0002 );
//clear
VmeCycle( SBS620_WORD, SBS620_WRITE, SBS620_AM_09, 0x22220006 );
```

```
// Initialize V550(ADC) for LAM"
//clear
VmeCycle( SBS620_WORD, SBS620_WRITE, SBS620_AM_09, 0x44440006 );
// set channel num : 768ch for 1 sensor
VmeCycle( SBS620_WORD, SBS620_WRITE, SBS620_AM_09, 0x44440004, 0x0618 );
// set channel num : 1536ch for 2 sensors
//  VmeCycle( SBS620_WORD, SBS620_WRITE, SBS620_AM_09, 0x22220004, 0x0c30 );

//int level 1
VmeCycle( SBS620_WORD, SBS620_WRITE, SBS620_AM_09, 0x44440000, 0x0101 );

//set pedestal & threshold 0
for( int add = 0; add<0x2000; add+=4 ){
  VmeCycle( SBS620_LONG, SBS620_WRITE, SBS620_AM_09,
            0x44442000 + add, 0x00000000 );
  VmeCycle( SBS620_LONG, SBS620_WRITE, SBS620_AM_09,
            0x44444000 + add, 0x00000000 );
}
//status reg 2
VmeCycle( SBS620_WORD, SBS620_WRITE, SBS620_AM_09, 0x44440002, 0x0002 );
//clear
VmeCycle( SBS620_WORD, SBS620_WRITE, SBS620_AM_09, 0x44440006 );
```

データ収集

VME サイクルでデータを読み込み、最後に V551B, 550 をリセットしてください。

```
// for read 1 sensor
for( i=0; i<768; i++ )   wdata = sbs620_read_data32( 0x22220008, SBS620_AM_09 );
// reset all modules
sbs620_write_data16( 0, 0x22220006, SBS620_AM_09 );
sbs620_write_data16( 0, 0x44440006, SBS620_AM_09 );
sbs620_write_data16( 0, 0x11110004, SBS620_AM_09 );
```

信号をオシロで見る～データ取得

*以下、シリコンの読み出しは 1 枚の例です。

オシロスコープで見る

図にセットアップのブロックダイアグラムを示します。トリガーとしては、ペデスタルを見るだけであればパルサー、ベータ線を見るようであれば、シンチをおいて、ベータ線の貫通イベントトリガーを作ってください。

シリコンのオペレーションなので、遮光が必要ですが、VA チップだけのオペレーションの際も遮光してください（Bare chip なので、VA チップが動かなかったりします)。また、経験上クーリングの必要はありません。

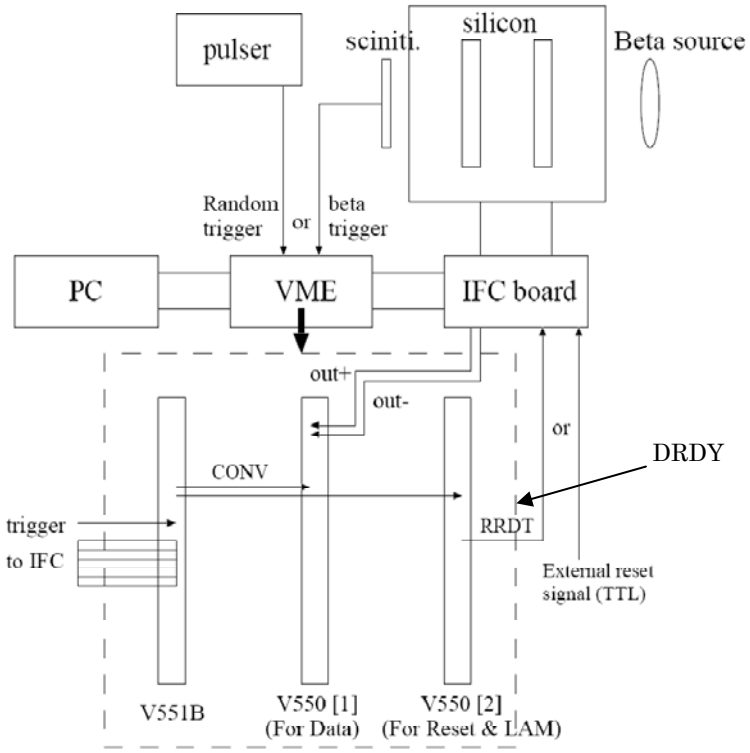
VA チップは V551B から N pulse 後に自動的に出力される、DRESET, ARESET によってリセットされますが(FIG.4.3 参照)、本セットではインターフェース上で物理的に切断しており、RESET シグナルは外部から入力しなければいけません（V551B の仕様で、N pulse が 2047 までしか設定できないが、それ以上を読みだす必要があったため)。RESET シグナルは LEMO で TTL 入力してください。以下の例では、この RESET シグナルを、読み出しとは別の V550 の DATA READY (DRDY)シグナルで行っています。DRDY は設定したチャンネル数分 ADC コンバージョンが起こり、メモリーにたまると TTL シグナルを出すので、これを全チャンネルのコンバージョンが終わったというシグナルに使用する。（しかしながら、DRDY に設定できるチャンネル数が 2016 までなので、3 台以上読む時はこの方法は使えません。その場合は OUTPUT register などで TTL シグナルを作り、データ読み出しの後などに入力してください)。RESET シグナルは図 6 の青い矢印で示したところに挿してください。

- 原理的にはいくつでもシリアルに読めるはずです。当初の実験では OUTPUT register を使って 7 台を読むことができました。シリアルだから時間かかりますけどね。

オシロで見る際は trigger (V551B に入れるもののコピー), convert, out+,out-（この３つ

は V550[1]に挿しているもの）シグナルを 4CH オシロで見てください。Trigger シグナルでトリガーし、convert に同期して、output が出てくるのがわかります。この時イベント毎にリセットがかかっていないと、はじめの一発だけ見えて、後は output が見えなくなります。RESET 用 V550 の設定（DRDY がきちんとトリガー毎に光っているか)、もしくは external reset の設定を確認してください。

うまく読み出せていれば図 9,10 のような波形が見えます。



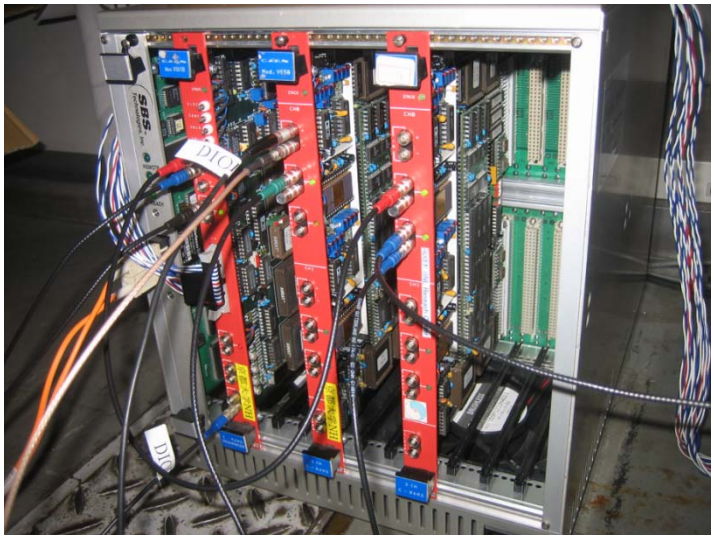


図８　上、セットアップ図。下、テスト時の VME セットアップ。

図 9　オシロで見たアウトプット。CONV シグナルと同期した OUTPUT が出ているのがわかる。各 OUTPUT の端は CLOCK の立下り（上がり）に同期してショット成分があるので、ここを外すように CONV とのタイミングを合わせる。V550 に入れる直前をみて、CONV の立下りが OUT の真ん中に来るように合わせる。

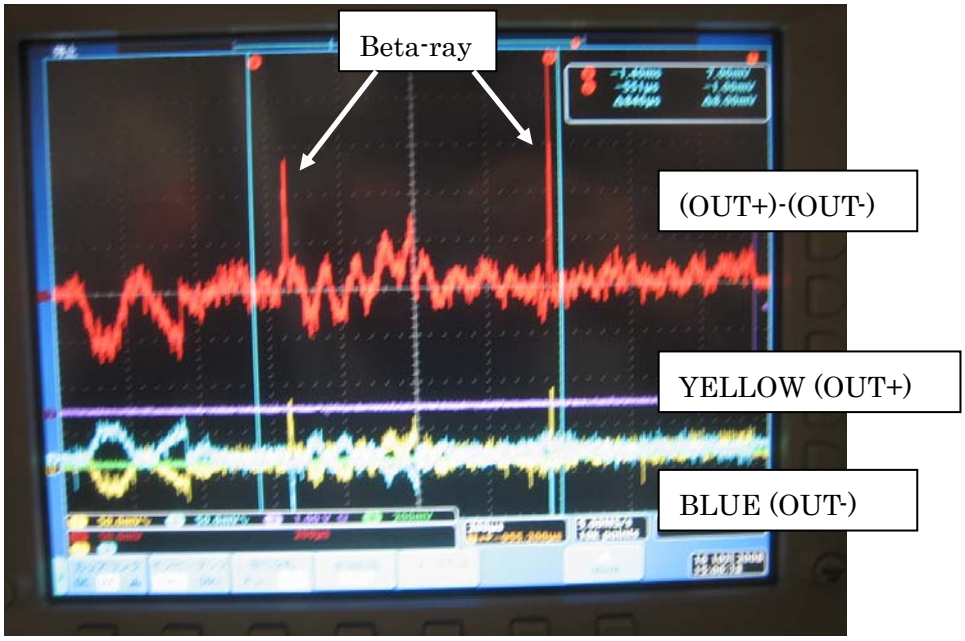


図 10　オシロで見たアウトプット。図 9 に対して横軸を拡大している。ベータ線源を置いてシンチレータでトリガーしている。ペデスタルのシェイプと、ベータ線源によるシグナルが見える。

ベータ線を見るのであればシリコンに電圧をかけてください(ORTEC710 を使用の場合、SHV-LEMO 変換をしますが、その際、抵抗などを挟んで下さい。用意する物参照)。HV の上げ方はやったことある人に聞いて下さい（PMT のように扱わないように。電圧の変化が電流をうみますので、電流値を見ながら、ゆっくり、連続的に上げてください。一気に uA とか飛び値するのは言語道断。その場では壊れないが、寿命に関係するそうです)。以下、typical な電流値です。50V くらいはかけたいと思うので、3,4 を使う際はもっと上げて結構です。電流値 10 uA までは流していいかと思います。(ブレイクダウンまで行ってたらダメです)

うまくいけば 1 shot のオシロの絵で、図 10 のような波形が見せます。

シリコン#	HV 値 (V)	漏れ電流 (μA)
3 (400 μm)	30.0	3.79
4 (400 μm)	30.0	3.89
9 (400 μm)	50.0	1.32
5 (250 μm)	60.0	2.84
6 (250 μm)	30.0	13.8
7 (250 μm)	60.0	2.7
8 (250 μm)	30.0	7.8

KEK テストの際にかけた電圧と、その時の漏れ電流の値。値は ORTEC710 の値

データ収集

オシロスコープで signal と convert タイミングを確認した後（図 9 参照)、それぞれを V550 に接続し、データ収集を行います。縦軸 ADC, 横軸 channel のスキャッタープロットを見ると、オシロと同じ情報になるので、その形が同形かみることによって正しくデータ収集ができているか確認できます。

図 11 にベータ線を置いた時のオンラインプロット、ADC vs. channel を載せます。オシロで見えていたような形のベースライン（ペデスタル）と、シグナルがちらほら見えます。これが見えたら勝ちです。

アナリシス

上記データ集出力をバイナリモードで見て、データ収集がうまくいっているとこまで確認しました（図 11)。アナライザーは DAQ の設計に合わせて各自作ってください。データフォーマットは V550 に書いてありますが、32bit にヘッダ、チャンネル、ADC の値が入っ

ています。例えば、

unsigned long FullData = buff;　(注：32bit)

int CH = (FullData&0x007ff000) >> 12;

int ADC = (FullData&0x00000fff);

のようなマスクをして CH, ADC 情報を取ってくればいいかと思います。

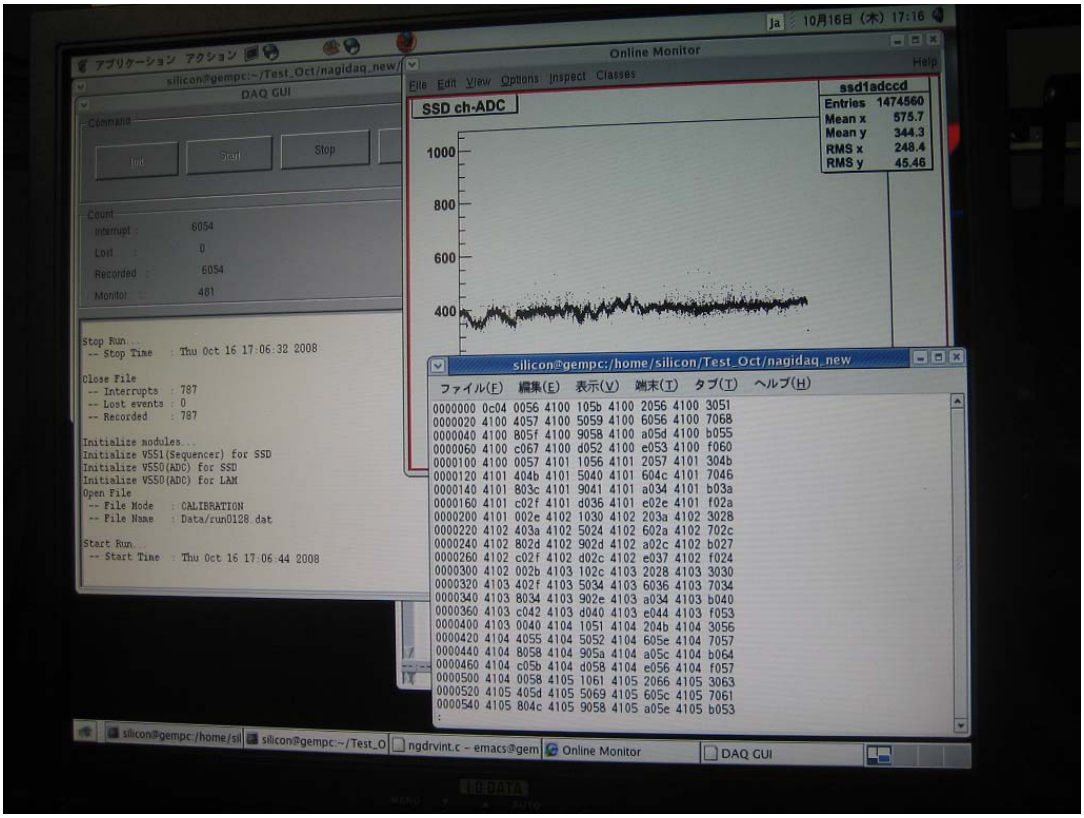


図 11　ベータ線の突き抜けトリガーを用いたデータ収集で、縦軸 ADC、横軸 channel でオンラインプロットしたもの。図 10 で見たようなペデスタルシェイプと、ベータ線のシグナルがちらほら見える。ターミナル内のバイナリデータは取得したデータを od で表示したもの。データストラクチャーは V550 仕様書参照。